INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
CEARA

DEPARTAMENTO DE TELEMÁTICA
DISCIPLINA: PROGRAMAÇÃO ORIENTADA A OBJETO
LISTA EXERCICIO

ALUNO: MARIA TERESA SALES MENDES **Data:30/09/2021**

1ª Questão (10 Escores). Associe a cada item da 2ª coluna um valor que corresponde a um item da 1ª coluna.

a)	Permite que um objeto seja usado no lugar de outro.	(C)	Encapsulamento
b)	Define a representação de um objeto.	(H)	Mensagem
c)	Separação de interface e implementação que permite que usuários de objetos possam utilizá-los sem conhecer detalhes de seu código.	(I)	Herança
d)	Possui tamanho fixo.	(A)	Polimorfismo
e)	Instância de uma classe.	(G)	Dependência
f)	Forma de relacionamento entre classes onde objetos são instanciados código.	(J)	Lista
g)	Forma de relacionamento entre classes implementado por meio de coleções.	(B)	Classe
h)	Forma de chamar um comportamento de um objeto.	(E)	Objeto
i)	Reuso de código na formação de hierarquias de classes.	(F)	Composição
j)	Permite inserções e remoções.	(D)	Array

2ª Questão (10 Escores). Aplique V para as afirmações verdadeiras e F para as afirmações falsas.

a) Métodos construtores devem sempre ser explícitos. (F)

b) A classe **Professor** tem um relacionamento de agregação com a classe **Disciplina.** (V)

c) Quando uma classe possui como atributo uma referência para um objeto temos uma dependência. (V)

d) Membros de classes static existem mesmo quando nenhum objeto dessa classe exista. (V)

e) Um relacionamento ´***tem um***´ é implementado via herança. (F)

f) Uma classe **Funcionário** tem um relacionamento ´**é um**´ com a classe **Dependente.** (F)

g) Uma classe abstract pode ser instanciada. (F)

h) Relacionamentos TODO-PARTE são tipos de associações. (V)

i) Você implementa uma interface ao subscrever apropriada e concretamente todos os métodos definidos pela interface. (V)

j) Um método ***static*** não é capaz de acessar uma variável de instância. (F)

3ª Questão (40 Escores). Escreva exemplos de código Python onde seja possível identificar os seguintes conceitos de POO.

a) Herança;

```
class pessoa:
    def _init_(self,nome,idade):
        self.nome=nome
        self.idade=idade

    def mostrarNome(self):
        print(self.nome)

    def mostrarIdade(self):
        print(self.idade)

class estudante(pessoa):
    def _init_(self,nome,idade,matricula):
        pessoa._init_(self,nome,idade)
        self.matricula=matricula

    def mostrarMatricula(self):
        print(self.matricula)
p=pessoa("marcos",30)
p.mostrarIdade()
s=estudante("isabel",20,100)
s.mostrarNome()
s.mostrarMatricula()
```

b) Encapsulamento;

```
class caixa:
    def _init_(self,altura,largura,comprimento):
        self._altura=altura
        self._largura=largura
        self._comprimento=comprimento
    def setAltura(self,valor):
        if str(valor).isnumeric():
            self._altura=valor
    def getAltura(self):
        return self._altura
    def volume(self):
        return self._altura*self._largura*self._comprimento
s = caixa(2,4,7)
print(caixa.volume)
```

c) Polimorfismo;

```
class apresUm:
    def ola(self):
        print("ola como vai?")
class apresDois:
    def ola(self):
        print("oi tudo bem?")
test=apresDois()
test.ola()
```

d) Variáveis de Instância;

```
class bankAcc():
    extrato=900
    def _init_(self,saldo):
        self._saldo=saldo
        bankAcc.extrato+=saldo
    def saque(self,v):
        self._saldo-=v
        bankAcc.extrato-=v
    def deposito(self,v):
        self._saldo+=v
        bankAcc.extrato+=v
    def getExtrato(self):
        return self._saldo
    def setSaldo(self,new):
        self._saldo=new
bankUm=bankAcc(900)
bankUm.deposito(8000)
print(bankUm.getExtrato())
bankUm.saque(400)
print(bankUm.getExtrato())
print(bankAcc.extrato)
```

e) Métodos construtores

```
class carro():
    def _init_(self,marca,ano,cor,uso):
        self.marca=marca
        self.ano=ano
        self.cor=cor
        self.uso=uso
car=("Ford",2014,"branco","seminovo")
```

f) Dependência

g) Associação

```
class pintura():
    def _init_(self,cod_obra,titulo):
        self.cod_obra=cod_obra
        self.titulo=titulo
class Pintor():
    def _init_(self,codPintor,nome,lugar):
        self.codPintor=codPintor
        self.nome=nome
        self.lugar=lugar
        self.obras=list()
    def getPintura(self,pintura):
        self.obras.append(pintura)
    def showPintura(self):
        for a in self.obras:
            print(a[1])
            print()
artVG=Pintor(1,"Van Gogh","Países Baixos")
obraUm=[1,"Starry Night"]
obraDois=[2,"Sunflowers"]
artVG.getPintura(obraUm)
artVG.getPintura(obraDois)
artVG.showPintura()
```

h) Relacionamento TODO-PARTE

```
class venda():
    def _init_(self,cod_venda,*produtos):
        self.cod_venda=cod_venda
        self.produtos=list(map(lambda x:x._dict_,produtos))
    def pay(self):
        for a in self.produtos:
            print(a["Nome"])
        print("total:",sum(list(map(lambda x : x["preco"], self.itens))))
class produto():
    def _init_(self,prod,valor):
        self.prod=prod
        self.valor=valor
vendaUm=venda(1,produto("banana",3.9),produto("feijao",7),produto("arroz",6.5))
vendaUm.pagamento()
```

4ª Questão (20 Escores)

Escreva em Python uma classe Ponto que possui os atributos inteiros x e y. Escreva uma classe Reta que possui dois pontos a e b. Escreva os métodos construtores para a classe Ponto e para a Classe Reta. Escreva os métodos get e set para acessar e alterar os atributos da classe Ponto e da classe Reta. Escreva um método distancia que retorna um valor real da distancia entre os dois pontos da reta.

```
import math
class ponto(object):
    def _init_(self,x,y):
        self.x=x
        self.y=y
    def getPonto(self):
        print(f'\nx: {self.x}\t y: {self.y}')
    def set_x(self,x):
        self.x=x
    def set_y(self,y):
      self.y=y

class reta(object):
    def _init_(self,ax,ay,bx,by):
       self.ax=ax
       self.ay=ay
       self.bx=bx
       self.by=by
    def getDistancia(self):
        d=math.sqrt((self.bx-self.ax)*(self.bx-self.ax)+(self.by -
self.ay)*(self.by - self.ay))
        print("a distancia e igual a {}".format(d))
    def setA(self,a):
        self.a=a
    def setB(self,b):
        self.b=b
if name=="main":
    x=int(input("insira um valor para x: "))
    y=int(input("insira um valor para y: "))
    print("Forme o ponto a com ax e ay")
    ax=int(input("insira um valor para ax: "))
    ay=int(input("insira um valor para ay: "))
    print("***")
    bx=int(input("insira um valor para bx: "))
    by=int(input("insira um valor para by: "))
    r=ponto(x, y)
    r.getPonto()
    s=reta(ax,ay,bx,by)
```

```
    s.getDistancia()
```